REMARKS

No claims have been cancelled or added. No claims have been amended. Claims 1-48 are currently pending in the application. In view of the following remarks, Applicant respectfully requests withdrawal of the rejections and forwarding of the application onto issuance.

The § 102 Rejections

Claims 1-10 and 25-48 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,757,919 to Herbert et al (hereinafter "Herbert").

The § 103 Rejections

Claims 11-18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Herbert in view of U.S. Patent No. 5,628,023 to Bryant et al. (hereinafter "Bryant").

Claims 19-24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Herbert in view of U.S. Patent No. 6,003,117 to Buer et al. (hereinafter "Buer").

Claims 1-10

Claim 1 recites, in a paging operating system having physical memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the physical memory, a computer-implemented method of protecting information comprising [emphasis added]:

- encrypting information using a ***key*** that is ***page-locked*** in the physical memory; and
- paging out, to the page file, the encrypted information.

In making out the rejection of this claim, the Office argues that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced below, in support of its argument.

> The page directory is similar to the page table except that it holds the base address of a number of page tables. Typically, it is retained in internal memory and not paged out. The base address of the page directory is held in an internal CPU control register. Functioning of paging systems is generally well understood in the art. *Col. 1, lines 35-40.*
>
> One embodiment of the invention employs a paging hierarchy in which the ***page directory*** always resides in the secure RAM 14 and a page table is associated with each application. The page table could be paged out when the application is not in use. For example, upon verification of the digital signature, the page table could itself be paged out just like any other page. The paging out process is discussed further in connection with FIG. 5 below. *Col. 5, lines 26-40.*

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it page locks a ***key*** used to encrypt information that is paged out from physical memory. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely contains base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are page-locked. Accordingly, for at least these reasons, this claim is allowable. If the Office insists upon maintaining this rejection, Applicant

respectfully requests that the Office specifically point out where Herbert teaches that the ***keys themselves*** are page-locked.

Claims 2-10 depend either directly or indirectly from claim 1 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 1, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

For example, **claim 3** recites that the creating of the key comprises creating the key during ***system boot up***.

The Office argues that Herbert teaches a method for creating the key during system boot up and cites to column 2, lines 45-52, reproduced below, for support:

> The flash memory is used for long-term storage of secret information, and a portion thereof may be allocated to applications running in the secure environment Additionally, in one embodiment, the real time kernel for secure processor 16 is stored in the flash memory. This allows all basic operations to be performed without external intervention (active or passive), and improves performance as compared to moving the kernel through the security services described herein.

Applicant respectfully submits that the excerpt cited by the Office does ***not*** teach that a key is created during ***system boot up***. Applicant directs the Office's attention to column 4, lines 7-25, and column 6, lines 59-60, which teach that Herbert's key is generated ***at the time of software installation***. Those excerpts are reproduced below [emphasis added]:

> At some time, software must be installed in the secure environment. Such "off the shelf" software will, of course, not be encrypted in the manner used within the secure environment. It will typically have a digital signature which can be used to verify the authenticity of the software being installed if digital signature verification is a supported function within the secure environment. ***FIG. 3 shows a flowchart of installation of a program in the secure system. At***

> ***functional block 120, a key is generated and initialization vector [IV] is generated for an application to be installed.*** Key generation can be accomplished using the random number generator which generates random bits. Random bits are collected until the desired key length is reached. In one embodiment, the random number generator has a thirty-two bit output register. The processor 16 reads the register a number of times necessary to collect enough random bits for a full key. Keys can be generated with one key for each application, i.e. all code pages and data pages associated with one application share the same key. *Col. 4, lines 7-25.*
>
> As discussed above, the encryption key and IV are generated ***at the time of installation.*** *Col. 6, lines 59-60.*

Applicant respectfully submits that Herbert ***specifically teaches*** that its key is generated ***at the time of software installation***. Accordingly, for at least this reason, this claim is allowable.

Claims 11-18

Claim 11 recites, in a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, a computer-implemented method of protecting information comprising [emphasis added]:

- ***page-locking*** a ***key*** in main memory;
- restricting access to the ***page-locked key*** to only the operating system kernel;
- calling the operating system kernel to encrypt information;
- accessing the ***page-locked key*** with the operating system kernel; and
- using the operating system kernel to encrypt the information with the ***page-locked key***.

In making out the rejection of this claim, the Office again relies on Herbert by arguing that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it page locks a ***key*** used to encrypt information. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are page-locked. Neither does the Bryant reference teach or suggest this feature. Accordingly, for at least these reasons, this claim is allowable. Again, if the Office insists upon maintaining this rejection, Applicant respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are page-locked.

Claims 12-18 depend either directly or indirectly from claim 11 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 11, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

Claims 19-24

Claim 19 recites, in a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, a computer-implemented method of handling encrypted information comprising [emphasis added]:

- accessing encrypted information in the page file; and

- decrypting the encrypted information with a ***key*** that is ***page-locked*** in the main memory.

In making out the rejection of this claim, the Office again relies on Herbert by arguing that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it page locks a ***key*** used to decrypt information. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are page-locked. Neither does the Buer reference teach or suggest this feature. Accordingly, for at least these reasons, this claim is allowable. Again, if the Office insists upon maintaining this rejection, Applicant respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are page-locked.

Claims 20-24 depend either directly or indirectly from claim 19 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 19, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

Claims 25-29

Claim 25 recites, in a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, a computer-implemented method of protecting information comprising [emphasis added]:

- allocating a non-pageable page of main memory;
- generating a random key; and
- storing the random ***key*** in the ***non-pageable page of main memory***, the random key being configured for use by the operating system to encrypt information that might be paged out to the page file.

In making out the rejection of this claim, the Office again argues that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that a random ***key*** is stored in a non-pageable page of main memory. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are stored in a non-pageable page of main memory. Accordingly, for at least these reasons, this claim is allowable. If the Office insists upon maintaining this rejection, Applicant

respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are stored in a non-pageable page of main memory.

Claims 26-29 depend either directly or indirectly from claim 25 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 25, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

For example, **claim 27** recites that the allocating takes place during ***system boot***.

The Office argues that Herbert teaches a method for creating the key during system boot up and cites to column 2, lines 45-52, reproduced above, for support. Applicant is unclear why the Office chooses to make this argument in reference to this claim. This claim recites ***allocating a non-pageable page of main memory*** at system boot. Although this claim recites that a random key is generated and stored in the non-pageable page of main memory, the generation and storage of the key do not necessarily take place at system boot. Applicant has reviewed the Herbert reference and respectfully submits that Herbert does not teach or suggest the subject matter of this claim. Accordingly, this claim is allowable. If the Office insists upon maintaining this rejection, Applicant respectfully requests the Office to specifically point out where Herbert teaches ***allocating a non-pageable page of main memory at system boot***.

Claims 30-35

Claim 30 recites, in an operating system having main memory for holding information and secondary storage for receiving information that is transferred out

of main memory, a computer-implemented method of protecting information comprising [emphasis added]:

- generating at least one ***non-pageable*** random ***key*** by using a random key generation process;
- encrypting at least one selected block of information in the main memory with a software component that uses the at least one random key for encryption;
- transferring the one encrypted block of information to the secondary storage;
- decrypting the one encrypted block of information with the software component that uses the at least one random key for decryption; and
- placing the decrypted block of information in the main memory.

In making out the rejection of this claim, the Office again appears to argue that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses a non-pageable random ***key***. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are non-pageable. Accordingly, for at least these reasons, this claim is allowable. Again, if the Office insists upon maintaining this rejection, Applicant respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are non-pageable.

Claims 31-35 depend either directly or indirectly from claim 30 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 30, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

For example, **claim 31** recites that the generating is performed during ***system boot up***.

The Office argues that Herbert teaches a method for creating the key during system boot up and cites to column 2, lines 45-52, reproduced above, for support.

Applicant respectfully submits that the excerpt cited by the Office does ***not*** teach that a key is generated during ***system boot up***. Applicant directs the Office's attention to column 4, lines 7-25, and column 6, lines 59-60, which teach that Herbert's key is generated ***at the time of software installation***. Those excerpts are reproduced below [emphasis added]:

> At some time, software must be installed in the secure environment. Such "off the shelf" software will, of course, not be encrypted in the manner used within the secure environment. It will typically have a digital signature which can be used to verify the authenticity of the software being installed if digital signature verification is a supported function within the secure environment. ***FIG. 3 shows a flowchart of installation of a program in the secure system. At functional block 120, a key is generated and initialization vector [IV] is generated for an application to be installed.*** Key generation can be accomplished using the random number generator which generates random bits. Random bits are collected until the desired key length is reached. In one embodiment, the random number generator has a thirty-two bit output register. The processor 16 reads the register a number of times necessary to collect enough random bits for a full key. Keys can be generated with one key for each application, i.e. all code pages and data pages associated with one application share the same key. *Col. 4, lines 7-25.*

> As discussed above, the encryption key and IV are generated ***at the time of installation.*** *Col. 6, lines 59-60.*

Applicant respectfully submits that Herbert ***specifically teaches*** that its key is generated ***at the time of software installation***. Accordingly, for at least this reason, this claim is allowable.

Claims 36-40

Claim 36 recites a system for use in protecting pageable information comprising [emphasis added]:

- a memory having pageable and non-pageable pages; and
- at least one ***key*** stored in the memory in a ***non-pageable page***, the key being configured for use in encrypting pageable information.

In making out the rejection of this claim, the Office again argues that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that a ***key*** is stored in the memory in a non-pageable page. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are stored in the memory in a non-pageable page. Accordingly, for at least these reasons, this claim is allowable. Again, if the Office insists upon maintaining this rejection, Applicant respectfully

requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are stored in a non-pageable page.

Claims 37-40 depend either directly or indirectly from claim 36 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 36, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

Claim 41

Claim 41 recites a computer program embodied on one or more computer-readable media, the program comprising [emphasis added]:

- encrypting information with a ***key*** that is ***page-locked*** in main memory of a computer;
- paging out, to secondary storage, the encrypted information;
- accessing the encrypted information in the secondary storage; and
- decrypting the encrypted information with the ***key*** that is ***page-locked*** in the main memory.

In making out the rejection of this claim, the Office again argues that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it page locks a ***key*** used to encrypt and decrypt information. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains

keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that its ***keys*** are page-locked. Accordingly, for at least these reasons, this claim is allowable. Again, Applicant respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are page-locked.

Claims 42-46

Claim 42 recites a programmable computer comprising [emphasis added]:

- a processor;
- main memory for holding information;
- secondary storage for receiving information that is temporarily transferred out of the main memory;
- the computer being programmed with computer-readable instructions which, when executed by the processor, cause the computer to:
  - encrypt information that is to be transferred to the secondary storage with a ***key*** that is ***locked*** in the main memory;
  - transfer the encrypted information to the secondary storage; and
  - decrypt the encrypted information with a ***key*** that is ***locked*** in the main memory.

In making out the rejection of this claim, the Office again argues that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it uses a ***key*** that is locked in main memory. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page

directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that it locks its ***keys*** in main memory. Accordingly, for at least these reasons, this claim is allowable. Again, if the Office insists upon maintaining this rejection, Applicant respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are locked in main memory.

Claims 43-46 depend either directly or indirectly from claim 42 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 42, are neither disclosed nor suggested by the references of record either singly or in combination with one another.

Claim 47

Claim 47 recites one or more ***application programming interfaces*** embodied on one or more computer-readable media for execution on a computer in conjunction with a paging operating system having main memory for holding information and a page file for receiving information that is paged out from the main memory, comprising [emphasis added]:

- an ***interface method*** for encrypting pageable information with a ***key*** that is ***page-locked*** in the main memory; and
- an ***interface method*** for decrypting encrypted information that is contained in the page file.

In making out the rejection of this claim, the Office argues that Herbert teaches an API comprising interface methods for encrypting and decrypting information as recited in Applicant's claim. The Office cites to column 2, lines 63-67, reproduced below, in support of its argument [emphasis added]:

> An encryption/decryption engine 12 encrypts outgoing pages and decrypts incoming pages at the ***interface*** before sending them to the external storage 4 or integrity check engine 13, respectively, thereby providing a confidentiality service.

The interface that Herbert refers to in the above excerpt is described at column 2, lines 41-43, reproduced below [emphasis added]:

> Bus 17 is electrically isolated from bus 7 by ***bus interface unit*** 19 which may be, for example, a bridge unit, and must be within the secure environment.

Applicant respectfully submits that the excerpt cited by the Office does ***not*** teach an ***application programming interface*** comprising ***interface methods*** for encrypting and decrypting information as recited in Applicant's claim.

The Office also again argues that Herbert discloses that the page containing key information is retained in the secure memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it uses a ***key*** that is page-locked in main memory. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn

contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that it page-locks its ***keys*** in main memory. If the Office insists upon maintaining this rejection, Applicant respectfully requests that the Office specifically point out where Herbert teaches that the ***keys*** are page-locked.

Accordingly, for at least these reasons, this claim is allowable.

Claim 48

Claim 48 recites an ***application programming interface*** embodied on a computer-readable medium for execution on a computer in conjunction with a paging operating system having main memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the main memory, comprising a ***method for setting an attribute*** on a page of main memory, the attribute designating that the page must be encrypted with a ***key*** that is ***page-locked*** in the main memory prior to the page being paged out to the page file.

Applicant again respectfully submits that Herbert does not teach an ***application programming interface***. Furthermore, Applicant has thoroughly reviewed the reference and can find no teaching of a method for ***setting an attribute*** which designates that a page must be encrypted with a ***page-locked*** key before being paged out. Applicant respectfully requests the Office to specifically point out where Herbert discloses setting such an attribute.

Also, in making out the rejection of this claim, the Office again argues that Herbert discloses that the page containing key information is retained in the secure

memory. The Office cites to column 1, lines 35-40, and column 5, lines 26-40, reproduced above, in support of its argument.

Applicant agrees that Herbert states that the ***page directory*** always resides in secure RAM. However, Herbert never discloses that it uses a ***key*** that is page-locked in main memory. Although the Office appears to argue that Herbert's page directory contains keys, Applicant strongly disagrees. Herbert defines what a page directory ***does*** in the excerpt reproduced above. Nowhere in the definition of "page directory" does Herbert teach or suggest that it contains keys. Rather, the page directory merely consists of base addresses of page tables, which in turn contain base addresses of page frames. Furthermore, Applicant has thoroughly reviewed the Herbert reference and respectfully submits that ***nowhere*** in the reference does Herbert disclose that it page-locks its ***keys*** in main memory. Accordingly, for at least these reasons, this claim is allowable. Again, Applicant respectfully requests the Office to specifically point out where Herbert teaches that the ***keys themselves*** are page-locked in main memory.

Conclusion

All of the claims are in condition for allowance. Accordingly, Applicant requests that a Notice of Allowability be issued forthwith. If the Office's next anticipated action is to be anything other than issuance of a Notice of Allowability, Applicant requests that the undersigned be contacted for the purpose of scheduling an interview.

Respectfully submitted,

Dated: 8/16/04

By: RR Cottle

Rob R. Cottle
Reg. No. 52,772
(509) 324-9256 ext. 247